

GÖTEBORGS UNIVERSITET

Humanistiska fakulteten

Översättarprogrammet

Institutionen för språk och litteraturer, källspråk engelska

Öppen översättning

Lokalisering av programvara med öppen källkod

Anders Warnqvist

Självständigt arbete, 15 högskolepoäng

Översättarutbildning 1, ÖU2100, Magisterutbildning

VT 2009

Handledare: Mats Mobärg

Examinator: Mats Mobärg

Sammandrag

Den här uppsatsen handlar om översättning av dataprogram med öppen källkod. En bakgrundsstudie om öppen källkod med inriktning på problematiken kring översättning, terminologi och språkvård genomförs. Eftersom öppen källkod delvis är ideologiserad i ett samhällsperspektiv, uppmärksammas också den ställning som öppen källkod har i samhället. Som en del av arbetet översattes det fria översättningsverktyget OmegaT till svenska, för att få exempel på problem som kan förekomma. Studien visar att den användarbaserade översättningen av öppen källkod inte nödvändigtvis leder till sämre översättningar, som varit en vanlig uppfattning. Tvärtom fungerar de grundläggande funktionerna inom öppen källkod för att även upprätthålla välgjorda översättningar, i den mån en aktiv användarbas finns på målspråket.

Nyckelord: Översättning, lokalisering, öppen källkod, fri programvara, språkvård, terminologi

Innehållsförteckning

1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte	2
1.3 Metod och material	2
2 Språk och öppen källkod	4
2.1 Lokalisering	4
2.2 Öppen källkod och fri programvara	5
2.2.1 Lokalisering av program med öppen källkod	7
2.3 Översättningsverktyg	8
2.4 Datarelaterad språkvård	9
2.4.1 Öppen språkvård	10
3 Översättning av OmegaT	12
3.1 OmegaT – det fria (GPL) översättningsverktyget	12
3.2 Lokalisering av OmegaT	13
3.3 Allmänna och egna problem vid lokalisering av OmegaT	13
3.3.1 Termer specifika för operativsystem	13
3.3.2 Fönster och paneler	14
3.3.3 Översättningstermer	15
3.3.4 Sammansättningar och versaler	17
3.3.5 Direkta överföringar	18
3.3.6 Påskägg	19
3.4 Lokaliseringsarbetet med OmegaT	21
3.4.1 Varför lokaliserar man OmegaT?	21
3.4.2 Terminologi	21
4 Slutsatser om lokaliseringen av OmegaT	23
5 Sammanfattning	25
Ordlista	26
Material- och litteraturförteckning	27

1 Inledning

Given enough eyeballs, all bugs are shallow. (Raymond 2001).

1.1 Bakgrund

Bakgrunden till ämnet för uppsatsen är att jag, alltsedan jag började använda mig mer och mer av datorer, fastnade för att använda just program med öppen källkod (eng: *open source*). Dessa så kallade fria program (mer om begreppen fri gentemot öppen källkod i avsnitt 2.2) har skapats av personer som bestämt sig för att fylla ett hål i utbudet av en viss programtyp, och distribuerat det fritt på internet till vem som helst, utan betalning. Med ”fri” menas dock inte bara ”gratis” och hobbyprogrammering. Också kommersiella företag utvecklar programvara med öppen källkod, och utvecklare av fri programvara ber ofta om donationer för att stödja arbetet (Orski 2007:35). I annat fall kan användare ”betala” för sig genom att själva bidra till programmet genom att skriva kod, hjälpdokument, eller som jag: översätta programmet till ett nytt språk för att fylla en lucka i utbudet.

Liksom användningen av programmen är fria, är källkoden, alltså programmeringskoden, också öppen. Det betyder att det går att ändra programmet för att passa in i andra situationer inom andra operativsystem och, som min studie kommer handla om, också olika språkversioner.

Dessa språkversioner kallas även *lokaliseringar*, och görs inte av betalda översättare, utan av vanliga användare, som däremot ibland är översättare. En förutsättning för utvecklingen av fri programvara är just att användare som känner att de har haft användning för programmet själva bidrar till utvecklingen. Detta sker i form av en löst sammansatt organisation, vanligtvis ledd av en utvecklingsgrupp. Utvecklingsgruppen delegerar, samordnar och stöder användarnas programmering eller översättning med hjälp av e-postlistor, forum och chattgrupper. Utan att vara finansierade av dataföretag klarar dessa användargrupper ofta av att lokalisera programvara till långt fler språk än kommersiella programutvecklare.

Den svenskspråkiga användargruppen på internet är särskilt aktiv inom den fria internetsfären. Fastän svenskan hamnar först på plats 89 i listan över de största språken i världen sett till antal talare (Grimes 1999) hamnar den exempelvis på plats 11 i fråga om antal artiklar på den fria encyklopedin Wikipedia (“List of Wikipedias - Meta”).

Svenska är dock ett språk som sällan saknas i de stora proprietära (copyright-skyddade) programmen, men mindre språk gör det ofta. Detta har lett till att myndigheter i vissa länder har valt att ge stöd till lokalisering av programvara med öppen källkod för att säkerställa språkens existens och möjlighet till utveckling inom datavärlden (Souphavanh & Karoonboonyanan 2005:17).

1.2 Syfte

Syftet med den här uppsatsen är att studera hur funktionen bakom öppen källkod, och hur den fungerar för lokalisering av programvaran. Som en del av det arbetet kommer jag lokalisera det fria översättningsverktyget OmegaT till svenska, för att få exempel på hur lokaliseringsarbetet kan gå till. I detta ingår bl.a. tillvägagångssätt och organisation av lokaliseringsarbetet, samt undersökning hur kvalitetssäkring av materialet går till i praktiken inom utvecklingen av öppen källkod.

1.3 Metod och material

Uppsatsen består av två huvuddelar. Dels den inledande diskussionen om lokalisering, öppen källkod och datorspråkvård, dels en presentation av översättningen som gjordes under studien. Översättningsarbetet gick ut på att lokalisera översättningsverktyget OmegaT till svenska. Lokaliseringen bestod av att översätta dels användargränssnittet i programmet (menyer, dialoger, knappar etc.), dels dokumentationen, alltså manualen, till programmet. Eftersom den samlade dokumentationen är tämligen stor (runt 20 000 ord) har jag istället valt att fokusera på översättningen av användargränssnittet och *Snabbstarten*, en kort introducerande guide till den huvudsakliga manualen på ungefär 1 500 ord. Snabbstarten är troligen skriven av en enskild person, till skillnad från den samlade manualen. Detta ger en mer homogen text, vilket lämpar sig bättre som exempel på en översättning av mer löpande text.

Då koden i sin helhet är alldeles för lång och komplicerad, passar den inte att presenteras i sin helhet som översättningsmaterial. Istället har

exempel ur materialet använts, som redovisas i kapitel 3. Programmet kommer dock att göras tillgängligt på svenska via OmegaT:s webbplats.

Allt originalmaterial som översattes finns dessutom på webbplatsen *SourceForge* (<http://www.sourceforge.net>), ett projekt som stöder program med öppen källkod genom att erbjuda nedladdningssidor och förvaring med långt större bandbredd än de webbplatser som de öppna programvarorna oftast har tillgång till. Större bandbredd betyder att fler kan ladda ner program samtidigt, något som annars kan få en mindre server att krascha (Holmevik 2004:148). Filerna som ska översättas kan sedan lätt öppnas i OmegaT självt eller i ett annat översättningsverktyg. Då programmeringskod för dataprogram egentligen består av vanlig text, skulle den därför kunna översättas i en vanlig textredigerare. Det skulle dock bli svårare att inte oavsiktligt skada programmeringskoden om man inte är van att hantera kod sedan tidigare. Ett översättningsverktyg underlättar detta avsevärt genom att bara presentera den text som faktiskt ska översättas (se avsnitt 2.3 nedan om översättningsverktyg).

Under översättningens gång har jag varit tvungen att konsultera och undersöka förekomsten av vissa termer och begrepp, då jag sedan tidigare inte har någon kunskap om programmering. De flesta termer har hittats i *Norstedts Stora Engelska Ordbok* och dataordlistor från svenska Datatermgruppen och finländska Forskningscentralen för de inhemska språken (Focis). I andra fall har sökandet lett mig in på internetforum och webbplatser om öppen källkod, som tas upp i avsnitt 2.4.1. Arbetet har också lett mig in på området datorspråkvård, som diskuterats flitigt inom svensk språkvård på senare år, främst i svenska Språkrådets och dess föregångare Språknämndens publikationer som *Språkvård*, *Språktidningen* och andra skrifter. Även uppsatser om datarelaterad översättning ur Göteborgs universitets serie *Texter emellan* har fungerat som inspiration och vägledning för utformning av uppsatsen och sökandet av sekundärlitteratur.

2 Språk och öppen källkod

Detta kapitel tar upp de grundläggande begreppen runt lokalisering, introduktion till vad öppen källkod är, och varför lokalisering av öppen källkod är intressant ur ett samhällsperspektiv. Dessutom finns en kort genomgång om vad ett översättningsverktyg gör samt arbete kring data-relaterad språkvård. Det är nödvändigt att gå igenom funktionen bakom utvecklingen av öppen källkod för att klargöra skillnaden mot kommersiella program.

2.1 Lokalisering

Med *lokalisering*, eller *l10n* som det kallas inom programmeringsjargongen, menas egentligen översättning av programvara eller webbplatser till ett annat språk eller för ett annat land (vilket givetvis kan ha samma språk, men inte samma stavning, kontaktinformation etc.). L10n står för begynnelse- och slutbokstaven samt antalet bokstäver däremellan i ordet *localization*. I18n, *internationalization* och g11n, *globalization*, har bildats på samma sätt och är någorlunda vedertagna termer inom programmering.

Lokalisering görs främst för att underlätta för ett program eller en webbplats att få fler användare, kunder eller besökare. Detta betyder heller inte alltid interlingval översättning av information till ett annat språk utan anpassning till andra format, marknader och kulturer (Nielsen 2001:315). Ingo beskriver detta som *adaption* när det gäller att anpassa en text genom att ta till en ny situation för att föra fram budskapet. Ingos beteckning urskiljer alltså också från enbart interlingval översättning av information (2007:153). IBM benämner däremot sina brittiska versioner som ”översättningar” (Kallin 1989:21). Vad gäller lokalisering av programvara är det främst användargränssnittet som översätts, men ibland också tillhörande manual eller webbplats om programmet. Lokalisering av programvara kräver ofta ett översättningsverktyg, men kan också göras i en vanlig textredigerare, förutsatt att man har tillräcklig kunskap om vad som är programmeringskod och vad som tillhör informationen i användargränssnittet.

En motpol till lokalisering är *internationalisering*, ibland även kallad *globalisering*. I dessa fall tillhandahålls ett program eller en webbplats på ett språk, fast för en global marknad, oftast på engelska (Nielsen 2001, s.315). Internationalisering av webbplatser och programvara är billigare och framförallt enklare än lokalisering, som kräver mer översättningsarbete. Fördelen med lokalisering är dock att den ger ett bättre intryck på en kund eller användare och ökar trovärdigheten om innehållet finns tillgängligt på det egna språket (A. Karlsson 2009:6).

Litteraturen om lokalisering finns främst tillgänglig om lokalisering av webbplatser. Principerna kan däremot ofta överföras till lokalisering av programvara, eftersom liknande krav på välstrukturerad information ställs i båda fallen. Datortexter, och därför också dess lokaliseringar, bör sträva efter *maximal användbarhet* på olika plan, då texten på de olika språken måste anpassas till den slutgiltiga användaren (A. Karlsson 2009:2). En välgjord lokalisering tar hänsyn till aspekter som grammatik, pragmatik, ekvivalens och layout med målet att uppnå denna maximala användbarhet.

2.2 Öppen källkod och fri programvara

Öppen källkod och fri programvara är i princip samma sak. De brukar t.o.m. sättas ihop under samlingsbeteckningen *Free/Open Source Software* (FOSS). Man skulle kunna säga att "öppen källkod" är den lite mer allmänna benämningen. Därför använder jag begreppet *öppen källkod* i den här uppsatsen för att beskriva utvecklingen av dessa program.

Förespråkare av de olika benämningarna fokuserar på olika fördelar med öppen källkod, och även om de gemensamma nämnarna är många finns det en viss konflikt mellan ideologi och pragmatik mellan de båda strömningarna (Holmevik 2004:125). Denna rörelse är delvis starkt ideologiserad vilket förstås färgar litteraturen som finns tillgänglig i ämnet. Förespråkare som Richard Stallman och Eric Raymond, som representerar de två huvudsakliga strömningarna i fråga om öppen källkod, har båda skrivit böcker som propagerar för just deras variant.

För att ett program ska räknas som "öppet" ska det licensieras med en licens godkänd av *Open Source Initiative* (OSI). OSI ställer krav på att programmet måste publicera hela sin källkod, att det ska vara fritt att använda i vilka situationer som helst och att programmet ska vara gratis.

Fri programvara är bl.a. licensierat under *GNU Public License* (GPL), som ställer högre krav på eventuella vidareutvecklingar av programmet,

men utvecklingen och organisationen av användargrupper är densamma som för program med öppen källkod.

Om ett program en gång har licensierats under GPL måste all vidareutveckling också distribueras under GPL. Detta för att säkerställa att utvecklingen som sker inom fri programmering inte kan köpas och beläggas med en betallicens. Program som licensierats genom OSI kan däremot införlivas i proprietära program. På senare tid har även andra tjänster licensierats under GNU-licenser, t.ex. den fria encyklopedin Wikipedia, men också e-böcker och manualer.

Utvecklingen av program med öppen källkod skiljer sig däremot markant från ”traditionell” programmering, då det är användarna som till stor del driver utvecklingen framåt genom nätverk som organiserar dessa aktiva användare att reagera på problem och brister. En av de främsta förespråkarna för öppen källkod, Eric Raymond (2001), delar in programutveckling i två olika varianter: *Katedralen* och *Basaren*.

Katedralen används oftast för proprietära program, men även vissa program med öppen källkod använder sig av detta arbetssätt. Här levereras programmet av dem som utvecklat det i ett färdigt paket. Om programmet har en öppen källkod, levereras källkoden med programmet, eller kan laddas ner separat från internet, om det handlar om ett program med öppen källkod. Ändringar sker när ledningen för utvecklingsgruppen har arbetat fram dem.

Basaren är en mer användarbaserad variant, där källkod och program ständigt är i utveckling, eftersom användare har full tillgång till koden under utvecklingsarbetet. De kan själva skriva till funktioner, tillägg, grafik eller för den delen lokaliseringar som tas upp i nästa uppdatering av programmet. Utvecklingen leds vanligtvis av en grupp frivilliga som samarbetar via internet, men som sällan eller aldrig träffas fysiskt. Eftersom användare uppmuntras att ta del i utvecklingsarbetet, antingen i form av att bidra med kod eller lokaliseringar eller bara testa hur programmet fungerar, upptäcks fel, så kallade *buggar*, snabbt och kan korrigeras. Detta arbetssätt upprätthålls enligt Raymond (2001) av *Linus lag*, döpt efter programmeraren Linus Torvalds, som skapade operativsystemet Linux. Denna lag lyder ”Given enough eyeballs, all bugs are shallow”. Basaren förutsätter att användare av naturen vill ha bra och säkra program, och i fallet med lokalisering även ett korrekt språk. Genom att tillåta dem att bidra och hjälpa till har programvara med öppen källkod blivit en succé. Både i Sverige och utomlands har öppen källkod fått alltmer utrymme i media, vilket lett till ökad användning (Orski 2007:57). Den ökade populariteten har också gjort att öppen käll-

kod har studerats i akademiska kontexter, om än med fokus på licensieringsfrågor.

2.2.1 Lokalisering av program med öppen källkod

Eftersom licenserna för program med öppen källkod är just ”öppna” är de också oerhört lätta att lokalisera. Eftersom källkoden, som släpps samtidigt med programmet oavsett om programvaran kommer från basaren eller katedralen, finns tillgänglig för alla kan vem som helst åta sig uppgiften att lokalisera programmet till ett annat språk om det inte skulle finnas tillgängligt innan. I princip kan översättaren släppa programmet på nytt, t.o.m. under nytt namn, bara han eller hon anger vem som skrivit koden innan, och var den kommer ifrån. Istället brukar de flesta utvecklare av programvara med öppen källkod ha en så kallad l10n-grupp som organiserar lokaliseringen av programmet. Detta är förstås praktiskt, eftersom två eller fler personer annars kan sitta på var sitt håll och lokalisera programmet, vilket vore onödigt. Den som är intresserad av att lokalisera ett program går helt enkelt med i l10n-gruppen, som vanligtvis kommunicerar med en e-postlista eller på ett forum. Där kan man få reda på om någon redan arbetar på en översättning, och hur man går tillväga för att lokalisera just det aktuella programmet.

Denna enkelhet, i samband med att det inte finns några proprietära licenser att ta hänsyn till, gör att program med öppen källkod är enklare att lokalisera än proprietära program och att de ofta blir översatta snabbare. Detta är något som uppmärksammas av flera tunga instanser vad gäller den öppna källkodens betydelse för datoranvändandet i utländer och bland äldre. Det är länge sedan svenska användare av de stora programmen behövde nöja sig med enbart en version på engelska. Svenskan är ett relativt stort språk, och en viktig marknad (Kallin 1989). Andra, betydligt mindre, språk får dock vänta på sina språkversioner, om de ens kommer. Lokala initiativ att lokalisera varumärkesskyddade program kan bestraffas hårt om de inte sanktionerats av utvecklaren. I en e-bok framtagen av *International Open Software Network* (IOSN), ett initiativ av FN:s utvecklingsprogram i Asien-Oceanien (UNDP-APDIP), framhävs att programvara med öppen källkod är att föredra just av denna anledning (Souphavanh & Karoonboonyanan 2005:9-12). Lokalisering av program med öppen källkod lyfts upp som en stor möjlighet att överbrygga de tekniska klyftorna mellan de rika och fattiga länderna. Öppen källkod är fri att lokaliseras, och det utan den extra kostnad som

licenser innebär. Ett problem med öppen översättning är dock att det, eftersom det saknas finansiell uppbackning från ett kapitalstarkt företag som vill få ut sin produkt på den nya marknad som ett nytt språk ger, kan vara svårt att få tillgång till professionella översättare att arbeta med det. Detta kan leda att de öppna översättningarna är av sämre kvalitet. Men eftersom lokaliseringarna också lätt kan uppdateras eller korrigeras av användare, i enlighet med *Linus lag*, motarbetas problemet. IOSN rekommenderar därför lokala utvecklingsprojekt att också arbeta med att lokalisera program och uppdatera tidigare lokaliseringar.

Lokala och myndighetsstödda lokaliseringsprojekt kan också hjälpa till att förbättra möjligheterna att använda små språk på datorn, något som annars skulle vara omöjligt. Detta kan i längden rädda små språk från att helt försvinna. För att underlätta arbetet med att lokalisera program med öppen källkod har UNDP-APDIP också tagit fram en guidebok om hur lokaliseringen av de mest använda programmen med öppen källkod går till, t.ex. kontorspaketet *Open Office* och webbläsaren *Mozilla Firefox* (Sasikumar m.fl. 2005). Precis som att öppen källkod anses vara viktigt ur demokratisynpunkt, genom att tillhandahålla program för användare som annars inte skulle kunna skaffa licenser för kommersiella program (Orski 2007:81), är lokaliseringar av dessa program lika viktiga för att fler ska kunna få användning för dem.

2.3 Översättningsverktyg

Ett *översättningsverktyg*, även kallat *CAT-verktyg* efter engelskans *CAT-tool (Computer Aided Translation)* är ett program som använder översättningsminnen för att underlätta översättning av texter med ofta förekommande uttryck, meningar eller termer. CAT ska egentligen särskiljas från *maskinöversättning*, även kallat *datoröversättning* eller *automatiserad översättning*, där översättningen sker helt utan mänsklig inblandning. Dessa termer blandas dock ihop, och det verkar finnas olika syn på uppdelningen. Ett exempel på maskinöversättning, som blivit populärt på senare tid är nättjänster som Google Translate (Austermühl 2001:154; Ahrenberg 1991; Grönroos 2005).

Ett översättningsverktyg importerar källdokument (som kan vara olika sorters textdokument), utvinnet endast texten och presenterar det för användaren. Programmet delar dessutom in texten i korta segment, eller översättningsenheter, antingen stycke- eller meningsvis. Medan användaren översätter segmenten till målspråket ett och ett, sparar översättningsminnet segmentparen. Om ett liknande segment återkommer, så

kommer översättningsverktyget att föreslå översättningen som gjorts innan. Dessa ungefärliga träffar kallas *luddiga träffar* (eng. *fuzzy matches*) och likheten anges i procent för att ge en fingervisning om hur bra den tidigare översättningen kan vara (Grönroos 2005). Om segmenten är helt identiska brukar översättningsverktyget vanligtvis automatiskt infoga den redan existerande översättningen. Allteftersom översättaren använder översättningsverktyget och fyller på databasen i översättningsminnet med nya segmentpar, kommer verktyget att kunna erbjuda fler luddiga träffar i fler situationer, vilket sparar tid för översättaren. Alternativt kan olika översättningsminnen användas till olika kunder eller ämnesområden, för att upprätthålla en konsekvent nivå på översättningarna (Austermühl 2001:134).

En annan fördel med att använda översättningsverktyg är de utvinnet endast texten som ska översättas, och kombinerar den översatta texten ur översättningsminnet för att exportera översatta dokument i samma formatering och samma filformat som man importerade. Alternativet vore att översätta direkt i dokumentet, med risk för att formatering eller disposition ändras. Detta är som tidigare nämnt tidsbesparande för översättare och därför kostnadsbesparande för uppdragsgivare. Detta har i sin tur orsakat lägre prissättning per ord för översättningar, eftersom det tar kortare tid att översätta åtminstone långa och upprepande texter (Lemhagen 1991). Det har också gjort marknaden för översättningsverktyg lukrativ, där mängder av översättningsverktyg finns tillgängliga.

Eftersom OmegaT är fritt att använda, både privat och kommersiellt, är det givetvis ett intressant projekt att arbeta med för att främja användandet av översättningsverktyg bland dem som inte har möjlighet att skaffa proprietära översättningsverktyg.

2.4 Datarelaterad språkvård

Datorspråkvård är av största vikt för att språket ska kunna uppvisa ett funktionellt, enhetligt datorspråk. I Sverige är Svenska datatermgruppen ledande för att skapa och upprätthålla en funktionell dataterminologi. Gruppen utgörs av representanter från Språkrådet, Terminologicentrum (TNC), Focis, press, universitet och dataföretag. I praktiken skapar oftast Microsofts och Apples egna översättare normen för det svenska dataordförrådet men påfallande många termer har antingen överförts direkt från engelska (t.ex. *router*; dessa direktöverföringar utgör ungefär en femtedel av de svenska vedertagna termerna), eller på engelska i en försvenskad form (t.ex. *skanner*). Detta fastän terminologiarbetet har en

uttalad princip om att bilda termer på begriplig och genomskinlig svenska (Kallin 1989). Men även ”vanliga” användare bidrar till många nya ord genom att de sprider nya termer på webbplatser, forum och nuförtiden även i allt större utsträckning på bloggar.

Språkvårdarna och även översättarna hänger inte alltid med i utvecklingen av nya termer, som så mycket annat inom datatekniken utvecklas fort. Även om översättarna för de stora dataföretagen till stor del är normbildande, betyder det inte att användarna följer normerna. Dessa terminologiska problem skapar förvirring hos dem som använder sig av datorer, däribland översättare som sysslar med lokalisering (O. Karlsson 1997:167; Mattsson 2003).

Språkvård inom datavärlden är också viktig ur demokratisk synvinkel. Många äldre i Sverige och i andra länder kan inte tillräckligt mycket engelska för att kunna förstå ett engelskt användargränssnitt, och ovana användare behöver genomskinliga termer på sitt eget språk för lättare kunna använda och förstå program. Denna aspekt tas också upp i lokaliseringsrapporten från IOSN (Souphavanh & Karoonboonyanan 2005).

2.4.1 Öppen språkvård

Öppen översättning, som jag har valt att kalla det i min uppsats, kan verka stå i motsatsställning till en effektiv och enhetlig språkvård, då enskilda användare kan ha begränsad kunskap om terminologi och för den delen också allmän språkkänsla.

I traditionell språkvård finns en försiktig skepsis till egentillverkade översättningar, och från Svenska datatermgruppen efterlyser man en högre uppskattning av språkvård samt att vedertagna termer bör främjas och användas i större utsträckning. Det finns en risk att dataspråket inte är tillgängligt för användare som inte behärskar jargongen som till viss del finns inom datatekniken.

Detta är däremot inte nödvändigtvis sant. Lokaliserare av program med öppen källkod lägger ner mycket tid och arbete på översättningen, och använder ofta dessutom programmet själva, vilket borde vara ett incitament att få till en så korrekt och användbar översättning som möjligt. Likaså finns bland öppna lokalisering ett aktivt engagemang för att översättningar ska bli korrekta och konsekventa. Att de som sysslar med datorprogrammering skulle vara ointresserade av språk och betydelsen av korrekta termer eller ”hätskt kritisera *löjliga* svenska data-termer” (O. Karlsson 1997) är inte intrycket man får när man besöker

lokaliseringsforum på internet. Det har upprättats omfångsrika ordlistor för termer i Linux, som används för att skapa enhetlighet i lokaliseringarna, då det är vanligt att flera personer delar på arbetet och sköter administration på forum och chattgrupper. Ett av dessa forum, *Translation Project* (<http://translationproject.org>) samordnar och främjar lokalisering av främst GNU-relaterad programvara. *Den svenska grenen av Translation Project* (<http://www.tp-sv.se>) länkar till flera projektspecifika termlistor skapade av vanliga användare som har arbetat med att lokalisera främst olika distributioner av Linux.

Translation Project granskar dessutom redan lokaliserade program och ser till att de håller en tillräckligt hög standard. Som allt annat inom öppen källkod är det upp till användare att peka på fel och brister och anmäla dem. Den ursprungliga utvecklaren av programmet förstår kanske inte svenska, och kan därför omöjligen granska lokaliseringen själv, men för att kunna nå ut till fler användare vill utvecklare gärna ha flerspråkiga program. Eftersom öppna licenser är oerhört flexibla kan däremot vem som helst som har anmärkningar på en lokalisering lämna in en ny för inkludering i nästa uppdatering. Eftersom användarna är organiserade i nätverk, fungerar granskningsprocessen genom *Linus lag*, för att ge en hög kvalitet även för lokaliseringar.

3 Översättning av OmegaT

Detta kapitel presenterar dels översättningsverktyget OmegaT och organisering kring dess lokalisering, dels specifika översättningsproblem som uppkommit under det egna översättningsarbetet. Dessa problem utgörs delvis av klassiska översättningsproblem mellan engelska och svenska, medan andra fall är mer specifika för lokalisering av programvara eller öppen källkod. Dessutom presenteras kommentarer ifrån andra som tidigare lokaliserat OmegaT till andra språk.

3.1 OmegaT – det fria (GPL) översättningsverktyget

OmegaT (<http://www.omegat.org>) skapades ursprungligen 2000 av Keith Godfrey, som försökte lansera programmet kommersiellt som ett lättviktsalternativ till andra översättningsverktyg på marknaden. När detta inte lyckades lade han istället ut programmet under den fria licensen GPL (se avsnitt 2.2 ovan) (Prior 2009).

OmegaT kan köras i flera olika operativsystem, t.ex. Linux eller Windows, eftersom det körs genom ett annat program, Java, som finns tillgängligt för de flesta operativsystem. Detta gör även programmet tillgängligt för de flesta datoranvändare, även i fattigare länder, som överlag har äldre datorer med sämre prestanda.

Projektet låg ett tag i malpåse men plockades upp av den nuvarande projektkoordinatör för OmegaT, Marc Prior. I princip är det projektkoordinatör som är ansvarig för utvecklingen, men arbetet utförs mestadels av en löst sammansatt arbetsgrupp bestående av användare i olika delar av världen (se avsnitt 2.2). Dessa har i sin tur ansvar för delområden som programmering, dokumentation (manualer och webbplats), lokalisering och användargruppen. De olika delansvariga har i stort total kontroll över sina respektive områden, och beslut tas av dem, snarare än projektkoordinatör eller utvecklingsgruppen. Beslut föregås oftast av diskussioner på olika forum inom användargruppen eller sändlistor (Prior 2009). Då OmegaT är licensierat under GPL, som definierar fri programvara, kallas det därför ”det fria översättningsverktyget”.

Det finns över 1 000 registrerade användare i användargruppen, och supporten sköts genom ett flerspråkigt forum där användare kan hjälpa varandra med problem och utbyta erfarenheter.

3.2 Lokalisering av OmegaT

OmegaT finns sedan tidigare lokaliserat till 29 språk. På OmegaT:s webbplats finns det en instruktionssida, en så kallad *howto*, där användare av programmet uppmuntras att bidra med sina översättningar av antingen OmegaT, dokumentationen eller del därav. Översättare kan samarbeta och ansvara för olika delar. Det finns ingen styrd, central kontroll av dessa lokaliseringar, utöver funktionskontroll för att kontrollera att programmeringskoden inte har skadats. Översättare har total frihet i hur de översätter. De kan modifiera, ändra, lägga till eller ta bort information som de önskar (Prior 2008).

Detta ses inte som ett problem, då den engelska originalversionen inte anses vara ett auktoritärt rättesnöre. Istället är det användbarheten som eftersöks, att översättare ska kunna arbeta i en programmiljö som är anpassad efter deras behov (Prior 2009). Detta överensstämmer med vad Karlsson (2009:2) benämner *maximal användbarhet*. I fallet med öppen källkod är det dock användarna som i slutändan ska bestämma hur programmet utformas. Detta gäller även språkversionerna. Lokaliseringar till vissa språk, bl.a. portugisiska, har gjorts tillsammans av en grupp, vilket kan hjälpa att säkerställa bra översättningar.

3.3 Allmänna och egna problem vid lokalisering av OmegaT

Nedan följer en presentation och analys av några av de problem som kan uppstå vid lokaliseringen av OmegaT.

3.3.1 *Termer specifika för operativsystem*

Eftersom OmegaT kan köras på vilket operativsystem som helst, måste termerna som används kunna förstås av användare av olika operativsystem. Även om Windows och MacOS är dominerande i vanliga fall, är det inte alltid så när det kommer till program med öppen källkod. Dessa program är oftast mer populära bland användare som också har Linux.

Ett problem för lokaliseringar av sådana program är att Windows, MacOS och Linux ibland har olika benämningar på samma företeelser.

Ett bra exempel är det som kallas *mapp* (eng: *folder*) på en Windows PC och Mac, vilket ofta kallas för *katalog* (eng: *directory*) eller *post* i Linuxdistributioner. Däremot förekommer *katalog* i både Windows och Mac som benämning på större mappstrukturer. Den uppdelningen använde jag med i min översättning, se tabell 1.

TABELL 1: *Användning av termerna katalog respektive mapp*

Some project directories seem to have been moved.	Några projektkataloger verkar att ha flyttats.
Glossary Directory	Mapp för ordlista

Eftersom jag själv arbetar i en MacOS-miljö tenderar jag möjligen att använda termer som är bekanta för mig. Ändå har jag aktivt strävat efter en bred tillämpning av termer för att göra lokaliseringen så genomskinlig och lättförståelig som möjligt, oavsett operativsystem. I installationsinstruktionerna lade jag exempelvis till ytterligare instruktioner för att underlätta för Mac-användare, eftersom jag själv saknat den informationen i instruktionen. Detta är en bra anledning att låta översättare lägga till information, något som inte annars brukar tillåtas. Översättarna är som nämnt också användare av programmet, och de har därför anledning att också bidra till information som de anser borde finnas.

3.3.2 Fönster och paneler

När OmegaT körs öppnas ett huvudfönster, som i sin tur är uppdelat i tre arbetsområden eller paneler. Dessa arbetsområden är de mest använda funktionerna, och deras funktioner tämligen uppenbara. Ändå måste namnen vara genomskinliga och uppenbara för användaren. Nielsen (2001:108) skriver visserligen om webbdesign, men principerna om att menyer och upplägg måste vara tydliga för att hitta det man vill går att överföra.

När det gäller menyer är det också viktigt att användaren dels förstår vad de olika funktionerna gör, dels att de är igenkännbara från andra program. De flesta menyrader och funktioner har ofta vedertagna namn och benämningar, och jag har i de flesta fall baserat mina översättningar

på dessa. Se dock avsnittet ovan om att operativsystem ibland använder olika benämningar på funktioner och företeelser.

TABELL 2: *Fönster och menyer*

Editor	Redigerare
Match Viewer	Träffgranskare
Glossary Viewer	Ordlistegranskare
Project	Projekt
Edit	Redigera
Go to	Gå till
View	Visa
Tools	Verktyg
Options	Alternativ
Help	Hjälp

Alla benämningar på menyfunktioner är vedertagna och plattformsoverskridande, dels från andra programmenyer, dels från övriga funktioner inom operativsystem.

De tre panelerna, *Redigeraren*, *Träffgranskaren* och *Ordlistegranskaren*, har bildats från andra program. ”Granskare” används i flera program, t.ex. visningsprogrammet *Förhandsvisning* i MacOS. Även ”överblick” används i andra program, men ”granskning” verkar beskriva funktionen bättre. ”Redigerare” finns det gott om, t.ex. i konstruktioner som *HTML-redigerare* eller *Textredigeraren*.

3.3.3 Översättningstermer

De översättningsspecifika termer som används handlar mest om tekniken för översättningsverktyg. Just det svenska ordet för *CAT-tool*, ’översättningsverktyg’, är lite problematiskt, då det blir så mycket längre (8 tecken på engelska, 20 på svenska). Speciellt i äldre texter på svenska kallas översättningsverktyg ibland också för *CAT-verktyg*, *CAT-tool* eller enbart *CAT*, (Lemhagen 1991) men *översättningsverktyg* är en mer genomskinlig term, och är numera den vanligaste benämningen. Det finns alltså ingen anledning att använda sig av äldre engelska termer när en bra svensk term finns tillgänglig. Viss risk finns dock att förväxla det

med *maskinöversättning* (se avsnitt 2.3 ovan), men för verksamma översättare är den risken däremot knappast stor.

Även om *översättningsminne*, på engelska *translation memory*, först verkar vara av liknande längd används ofta förkortningen *TM* på engelska. Detta är mer genomskinligt och dessutom allmänt vedertaget till skillnad från en eventuell svensk förkortning *ÖM*, då filerna för översättningsminnen också kallas *TMX* (*Translation Memory eXchange*). Dock kallas översättningsminnen ibland för *TM* i äldre svensk facklitteratur inom översättning. Precis som för termen *översättningsverktyg*, finns heller ingen anledning att använda gamla engelska termer som *TM*. Under översättningen uppkom inga svårigheter med att använda *översättningsminne* vad gäller utrymme.

Benämningar som *källspråk* och *målspråk* för *source* och *target language* är förstås allmänt vedertagna. Däremot är källtexten inte alltid konsekvent i användningen av dessa termer. Detta tyder på att det engelska originalmaterialet inte nödvändigtvis är helt felfritt eller enhetligt. Detta kan vara ett exempel på problematiken med öppen källkod, nämligen att materialet kanske inte genomgår samma granskningsprocess som kommersiella programutvecklare har tillgång till. Å andra sidan förekommer buggar och misstag även i proprietära program.

TABELL 3: *Exempel på översättningstermer*

OmegaT is a free and open source multiplatform Computer Assisted Translation tool with fuzzy matching, translation memory, keyword search, glossaries, and translation leveraging into updated projects.	OmegaT är ett fritt, plattformsöverspännande översättningsverktyg med öppen källkod. Funktioner innefattar bl.a. luddiga träffar, översättningsminne, nyckelordssökning, ordlistor och översättningsflytt till uppdaterade projekt.
Searches are performed both in the source and the translation.	Sökningar utförs både i käll- och måltext.
You can simply accept the location of the subfolders, but make sure first that the source and translation language codes are correct.	Du kan helt enkelt godkänna placeringen av undermapparna, men se först till att språkkoderna för käll- och målspråk är korrekta.
Fuzzy matches	Luddiga träffar

Fuzzy matches är en översättningsteknisk term som finns i de flesta översättningsverktyg. Funktionen är att programmet föreslår uttryck eller meningar som översättaren använt tidigare, antingen i samma text eller tidigare översättningar (se avsnitt 2.3 ovan). Dessa träffar från översättningsminnet presenteras som ”fuzzy”, alltså att de mer eller

mindre stämmer överens med segmentet som ska översättas. I de verktyg som finns lokaliserade till svenska kallas funktionen fortfarande för ”fuzzy match-teknik” eller liknande (Grönroos 2005). Detta har jag dock valt att kalla för *luddiga träffar* i min översättning. Anledningen är att jag tycker att termer bör vara på svenska om möjligt, vilket förenklar för nya användare. Jag tror dessutom att betydelsen är genomskinlig och tämligen uppenbar även för en översättare som tidigare använt ”fuzzy match-teknik” i ett annat översättningsverktyg.

Då det verkar vara få översättningsverktyg som faktiskt finns lokaliserade till svenska tror jag dessutom inte att termen ”fuzzy match” är fullt vedertagen. När Grönroos (2005) skriver om funktionen sätts den dessutom inom citationstecken.

3.3.4 Sammansättningar och versaler

Sammansatta ord skrivs i svenskan i ett grafiskt ord, helst utan bindestreck. I engelskan däremot är *compound nouns* vanligen särskrivna och ibland sammanfogade med bindestreck. I lokaliseringen av programvara finns det många termer som måste sättas ihop. Vissa fungerar bra, andra blir inte lika lyckade. Vissa sammansättningar kräver dessutom så kallade bindesuffix mellan leden (t.ex -s- eller -e-). Det är dock inte alltid man kan vara konsekvent eftersom vissa situationer kräver en omskrivning snarare än en sammansättning.

TABELL 4: *Exempel på sammansättningar*

Keyword searches examine the current project and return all segments which contain all of the query words, in any order.	Nyckelordssökningar undersöker det aktuella projektet och återger alla segment som innehåller alla frågeord i fri ordning.
Exact searches look for the specified query string within the current project or in any directory or directory tree.	Exakta sökningar letar efter den angivna frågesträngen inom det aktuella projektet eller i en annan mapp eller mappträd.

I engelsk text har namn på funktioner och annat ofta stor begynnelsebokstav. Detta är inte lika accepterat i svenskan, även om detta på senare tid blivit allt vanligare, speciellt inom områden med stort engelskt inflytande, som datorprogrammeringen ändå är. Däremot kan stor begynnelsebokstav vara bra för förståelsen av att det är en speciell funktion som omtalas.

TABELL 5: *Versal i början av substantiv*

Create a New Project	Skapa ett nytt projekt
Editing Behaviour Options	Alternativ för Redigeringsbeteende
On the left is the Editor where you type your translation.	Till vänster är Redigeraren, där du skriver din översättning.

I programmet finns bara en sorts projekt, som bara kan skapas på ett sätt, genom att välja *Arkiv: Skapa nytt projekt*. Ordet *redigera* förekommer dock på olika ställen i menyalternativen, men i det andra exemplet menas inställningar för funktionen *Redigeringsbeteende*, vilket jag därför markerat med versal. Det tredje exemplet behandlar panelen *Redigeraren* som namnet på en funktion. Detta har också gjorts för att förtydliga för användaren. Ibland finns dessutom program i vissa operativsystem som kallas "Textredigeraren" eller enbart "Redigeraren".

3.3.5 Direkta överföringar

Som tidigare nämnts finns många datorrelaterade termer inte på svenska. Istället har jag använt den engelska termen, ibland med viss specificering. Som vid sammansättningar är problemet delvis att medan engelskan gärna bygger sammansatta termer utan att vara grafiskt sammansatta, ska dessa ord i svenskan även vara grafiskt sammansatta. Metoden att binda ihop med bindestreck används, främst i tillfälliga konstruktioner med egennamn enligt *Svenska skrivregler* (2000).

TABELL 6: *Exempel på direkta överföringar*

HTML Help Compiler files	HTML Help Compiler-filer
Arrays must not be null	Array får inte vara null
Exception while parsing:	Undantag under parsning:
Java(TM) Resource Bundles	Java(TM) Resource Bundles
How to handle tagged text	Att hantera taggad text

Detta handlar alltså oftast om namnet på filtyper, där namnen kan vara förkortningar eller beskrivningar, som inte fått svenska vedertagna begrepp. Enligt Ingo (2007:37) är egennamn ett av få fall då en direkt överföring är lämplig. Filtyper kan ses som namn, men det är givetvis

bäst om de får namn på svenska. De fall som jag behållit engelska termer delvis eller i sin helhet har jag först sett till att de faktiskt används i svensk text. Jag använde mig av både Google och Wikipedia för att undersöka detta. Även om dessa tjänster inte alltid ger en rättvis bild av användningen av enskilda ord och fraser, ger de en fingervisning om huruvida termerna faktiskt används i datasammanhang.

För det första och fjärde exemplet i tabell 6 finns vissa varianter på svenska, men de låter nästan lika engelska, vilket knappast bidrar till förståelsen. Om en användare skulle vilja söka efter termen på internet skulle det i så fall vara bättre att lämna den på engelska, då en sådan sökning kommer ge ett bättre sökresultat.

I femte exemplet använder jag *taggad text* istället för *markerad* eller liknande, då *tagg* anses vara vedertagen som term av Svenska datatermgruppen (2007).

3.3.6 Påskägg

Den största översättningsutmaningen under arbetet var intressant nog de delar som vanligtvis inte visas om programmet körs normalt. *Påskägg* (eng: *Easter Eggs*) är små undangömda funktioner eller dialoger i programvara som bara kommer fram om man trycker på vissa tangenter i en viss ordning eller utför ett visst kommando (D. Wolf & A. Wolf 2001). Dessa har alltså ingen egentlig funktion i programmet som sådant. De finns i många sorters programvara, allt ifrån Microsoft Word till tv-spel, men är särskilt populära bland program med öppen källkod. Fenomenet kommer från den främst amerikanska traditionen att leta efter undangömda godisägg på påskdagen. På samma sätt måste man leta efter påskägg i program, då de inte kommer fram normalt sett.

De påskägg i OmegaT som jag hittade i koden för användargränssnittet blev ett intressant problem, då de är smått poetiska i sin karaktär. Jag hade dessutom ingen aning om att de fanns i programmet innan jag kom fram till den delen av koden.

TABELL 7: Påskägg

Exciting help page. Gossamer threads hold you back. 404 not found	Spännande hjälpsida. Du hålls tillbaka av spindelväv. 404 finns inte
Bartender yells loud: Your page cannot be found, boy. Buy another drink.	Bartendern ropar högt: Din sida kan inte hittas, grabben. Köp en till drink.
Page slips through finge. Pulse pounding hard and frantic. Vanishing like mist.	Sidan glider mellan fingrarna. Pulsen slår hårt. Försvinner spårlöst.
These three are certain: Death, taxes, and page not found. You, victim of one.	Tre saker i livet är säkert: Döden, skatter och sidan hittas ej. Du: ett offer för en av dem.
Ephemeral page. I am the Blue Screen of Death. No one hears your screams.	Kortlivad sida. Jag är dödens blå skärm. Ingen hör dina skrik.
Rather than beep. Or a rude error message. These words: 'Page not found'	Istället för ett pip. Eller ohövlit felmeddelande. Dessa ord: "Sidan kan inte hittas"
Help you are seeking. From your path it is fleeing. Its winter has come	Hjälp söker du. Ur din väg flyr den. Dess vinter har kommit

Först ville jag nästan behålla dem på engelska, då de egentligen inte tillhör det grafiska användargränssnittet, men tyckte sedan att det skulle vara roligt att översätta dessa små skönlitterära texter.

I de här exemplen kommer Eugene Nidas (1964:159) teorier om *formell* och *dynamisk ekvivalens* till användning. Formell ekvivalens fokuserar på budskapet i texten, och att det är begreppen som ska överföras till läsaren, till skillnad från en dynamisk ekvivalens, där effekten av helheten ska bevaras. Eftersom dessa gåtfulla påskägg egentligen inte har någon funktion eller effekt som sådant, är en formell, nästintill bokstavlig ekvivalens att föredra. Detta gäller främst det sista exemplet, där det är den underliga ordföljden som poängteras. Möjligen refererar textstycket till Jedimästaren Yoda från Star Wars-filmerna, som använder sig av en ovanlig ordföljd. Om textstycket översatts till korrekt svensk ordföljd hade denna referens försvunnit, och stycket mist sitt budskap i form av denna populärkulturella referens.

I övrig text är det snarare funktionen och användbarheten som är det viktiga, som givetvis inte har någon betydelse när det kommer till påskäggen.

3.4 Lokaliseringsarbetet med OmegaT

Som en del av arbetet med den här uppsatsen skickade jag ut en förfrågan på OmegaT:s användarforum till lokalisere som ville bidra med sina kommentarer. Jag skickade också ut en förfrågan till eventuella svenska användare av OmegaT, men fick inga svar, vilket jag antar betyder att antalet svenskspråkiga aktiva användare (som är medlemmar i användargruppen) är mycket lågt.

I förfrågan eftersökte jag varför man väljer att lägga ner tid och arbete på att lokalisera OmegaT, reflektioner om arbetet, problem med terminologin och andra problem man kan stöta på.

Fyra översättare svarade. De hade översatt programmet och/eller manualen till brasiliansk portugisiska, franska respektive ryska.

3.4.1 *Varför lokaliserar man OmegaT?*

Alla nämnde att de fastnat för programmet när de använt det för översättningsuppdrag, och beslutat sig för att göra det tillgängligt på sitt eget språk.

Den ryske lokalisaren undervisade i översättning på ett universitet och lokaliserade programmet för att kunna använda det i undervisningen. En av de franska lokaliserna nämnde att han ansåg arbetet vara betalning för att använda programmet. Han var även aktiv i andra lokaliseringprojekt och hade initierat och samordnat den japanska lokaliseringen av OmegaT. Den andre franske lokalisaren tyckte att ett engelskt användargränssnitt var "distraherande" när man arbetade på franska. Han uppgav dessutom att det dåvarande franska användargränssnittet var dåligt och att det behövde uppdateras. Han är alltså ett exempel på någon som inte varit nöjd med en existerande översättning, och gjort en egen, förbättrad version. Den brasilianska översättaren ville kunna bidra till att kollegor i Brasilien skulle kunna få tillgång till OmegaT lättare. Även om många där har varierande kunskaper i engelska, är det givetvis lättare att använda ett verktyg till fullo på sitt eget språk.

3.4.2 *Terminologi*

Det kan vara värt att påminna om att de flesta som använder (och lokaliserar) OmegaT är professionella översättare. Vissa har kanske

lokaliserat andra program tidigare, och är vana med terminologin som används. Den ryske lokalisaren, som även undervisade i översättning, kände sig helt säker på terminologin, medan den franske hade gjort arbetet lite amatörmässigt efter eget huvud. Det var hans version som senare blev korrigerad av en annan användare. Den andre franske lokalisaren nämnde normala tekniska ordböcker som källor, men även specifika ordlistor för Microsoft, Novell och linuxdistributionen Gnome. Då OmegaT är skapat för alla operativsystem måste även terminologin anpassas för att passa in i alla operativsystem. Den portugisiska lokaliseringen gjordes tillsammans av en grupp, som hjälptes åt med terminologi. Översättaren jag hade kontakt med hade tidigare lokaliserat programvara. Det är dock värt att notera att två lokaliseringar finns: en på brasiliansk portugisiska och en på europeisk portugisiska. Huruvida terminologin skiljer sig mellan de båda formerna av portugisiska fick jag inte reda på. Det var endast den portugisiska lokaliseringen som hade gjorts tillsammans av en grupp av översättare, i de andra fallen var det endast en person som gjort lokaliseringen.

Ett exempel på att användare och lokalisare är intresserade av språkvårdsfrågor är diskussion på användarforumet huruvida mappar ska kallas *folder*, 'mapp', eller *directory*, 'katalog', i det engelska användargränssnittet (se avsnitt 3.2 ovan). I slutändan spelar dessa små skillnader ingen större roll för funktionen, men belyser betydelsen som språket och användarnas reaktioner ändå har fråga om programmering av öppen källkod (Helary m.fl. 2009).

4 Slutsatser om lokaliseringen av OmegaT

Då OmegaT är ett översättningsverktyg som används av översättare är det naturligt att användare är intresserade av att lokalisera det till sina respektive språk. Programmet strävar efter att vara så tillgängligt som möjligt för en bred, internationell användarkrets.

Lokaliseringen av OmegaT till svenska bjöd på intressanta problem, som går utanför den vanliga översättningsproblematiken. Lokalisering innebär ofta anpassning till ett existerande fysiskt format som måste tillgodoses, samtidigt måste vedertagna termer användas i största möjliga mån. Så kallade direktöverföringar där engelska behålls bör helt undvikas av flera skäl. Främst för att främja förståelsen och användbarheten, men även för att stärka svensk terminologi på området. Dock förekommer direkta överföringar även i min översättning. Då handlar det främst om namn på dokumenttyper, men även andra helt vedertagna engelska termer eller försvenskade engelska termer.

Bland användare av programvara med öppen källkod finns det givetvis ett intresse av att ha bra, välfungerande program. Detta gäller även lokaliseringar. Det verkar dock inte anses att kvaliteten på lokaliseringarna skulle vara ett problem för utvecklare av programvara med öppen källkod. Projektkoordinatorn för OmegaT hänvisade till att de som använder programmet vanligtvis är översättare, samt att den engelska versionen inte ska anses som ett rättesnöre. Detta kan verka konstigt, men projektkoordinatorn är själv översättare och inser därför kanske att vissa skillnader måste accepteras. Användargränssnittet och dokumentation måste sträva efter en maximal användbarhet för sitt språk, och det fanns dessutom fall där den engelska källtexten inte var fullt enhetlig eller konsekvent. Därför ges översättaren av OmegaT full frihet i sin översättning. Denna frihet ger också möjlighet för översättaren att själv bidra med information som denne tycker saknas eller inte är tillräckligt tydlig, vilket främjar användbarheten av programmet. Denna inställning torde vara jämförbar med andra projekt med öppen källkod.

Översättningen av påskäggen blev en intressant utmaning, som gjorde att en helt annan översättningsmetod användes. Medan resten av arbetet fokuserar på användbarhet och funktionalitet på det nya språket, gällde det att bibehålla budskapet och kulturella referenser på olika plan.

Under mitt arbete har jag också tagit kontakt med andra som lokaliserat OmegaT, som till stor del styrker mina egna uppfattningar om program med öppen källkod i allmänhet och lokalisering av dessa i synnerhet. Alla dessa personer har känt att det funnits ett behov att översätta programmet till sitt eget språk, och velat göra det gratis som en sorts betalning för ett bra skrivet program. Terminologin som behövs kan hittas dels via traditionella ordböcker, ordlistor på internet samt förfrågningar via forum och e-postlistor.

OmegaT:s egen e-postlista för lokalisering används också för att diskutera frågor angående nya språkversioner, terminologi och översättning i allmänhet. Tyvärr har inte jag kunnat få kontakt med andra svenskspråkiga användare av OmegaT, något som jag hoppas kommer att ändras när programmet väl läggs ut på svenska. Andra forum för lokalisering har dock visat att det finns ett stort intresse från även svenska användare att engagera sig i de program de använder sig av. Dessa forum kan vara mycket användbara för lokalisering, då program ofta egentligen har liknande terminologi i användargränssnittet. Arbetet med att lokalisera det fria översättningsverktyget OmegaT har hittills gjort det tillgängligt på 29 språk. Inom OmegaT:s användargrupp finns ett aktivt lokaliseringsarbete inte bara organiserar arbetet, men också diskuterar och uppdaterar existerande versioner.

Dock behövs en större svenskspråkig användarbas för att funktionerna för *Linus lag* ska gälla den föreliggande svenska lokaliseringen.

5 Sammanfattning

Lokalisering av dataprogram har flera fördelar. Förutom att program på användarens eget språk är lättare att förstå och använda, är det viktigt ur ett samhällsperspektiv att ha en enhetlig dataterminologi på de inhemska språken. Att göra program tillgängliga på mindre språk stärker språkens ställning och motverkar språkdöd. FN:s utvecklingsprogram förordar att lokalisering av öppen källkod borde vara del av fler utvecklingsprojekt.

Lokaliseringen av programvara med öppen källkod har helt andra förutsättningar än av proprietär programvara. Eftersom licensieringen av öppen källkod är öppen för vem som helst, finns det risk för att dessa lokaliseringar skulle vara av lägre kvalitet. Men eftersom den princip som styr utvecklingen av öppen källkod, varigenom enskilda användare har möjlighet att rapportera fel och buggar för att gemensamt förbättra och utveckla programmet, även gäller för lokaliseringar, upprätthålls ett enhetligt språk och en maximal användbarhet även inom öppen översättning. Vidare finns även externa projekt inom öppen källkod som *Translation Project* som arbetar med en översyn av redan existerande lokaliseringar och främjandet av nya.

Mitt eget översättningsarbete med OmegaT har visat att organisationen för lokalisering fungerar även i praktiken. Trots att medlemmar i användargruppen kanske inte behärskar varandras språk finns en aktiv diskussion på forumen, hur olika lokaliseringar ska organiseras och uppdateras. Utvecklingen för respektive språk beror förstås på antalet användare med det aktuella språket. Arbetet visar också att det är relativt enkelt att lokalisera programvara med öppen källkod, samt att det finns terminologi tillgänglig från traditionella och internetbaserade källor, som möjliggör användarbaserad lokalisering.

Ordlista

CAT – Computer Aided Translation

FSF – Free Software Foundation

G11n – Globalisering

GNU – GNU's Not Unix

GPL – GNU Public License

HTML – HyperText Markup Language (används för att skriva kod för webbinnehåll)

Focus – [Finländska] Forskningscentralen för de inhemska språken

I18n – Internationalisering

IOSN – International Open Software Network

L10n – Lokalisering

OSI – Open Source Initiative

proprietär – copyright-skyddad

TNC – Tekniska nomenklaturcentralen (Terminologicentrum)

UNDP-APDIP – United Nations Development Programme - Asia-Pacific Development Information Programme

Material- och litteraturförteckning

Primärmaterial

OmegaT 1.8.1 lokaliseringspaket. Hämtat från:
http://sourceforge.net/project/showfiles.php?group_id=68187&package_id=69854 [Åtkomstdatum 12 februari 2009].

Sekundärlitteratur

- Ahrenberg, L., 1991. Maskinöversättning – en kort översikt. I:
Språkvård 1991:1.
- Austermühl, F., 2001. *Electronic Tools for Translators*. Manchester: St. Jerome Publishing.
- Grimes, B.F., 1999. Top 100 Languages by Population (uppdaterad lista från *Ethnologue* 13). Hämtad från:
<http://web.archive.org/web/19990422030645/www.sil.org/ethnologue/top100.html> [Åtkomstdatum 5 maj 2009].
- Grönroos, M., 2005. Från översättningsminne till översättningsintelligens. I: *Språkbruk* 2005:4.
- Helary, J., 2009. OmegaT : Message: Tidying up the English GUI !
OmegaT User group.
 <<http://tech.groups.yahoo.com/group/OmegaT/message/13402>>
 [Åtkomstdatum 28 april 2009].
- Holmevik, J.R., 2004. *TraceBack: MOO, Open Source, and the Humanities*. Bergen: Universitetet i Bergen.
- Ingo, R., 2007. *Konsten att översätta. Översättandets praktik och didaktik*, Lund: Studentlitteratur.
- Kallin, S., 1989. IBMs insatser för språkvård och klar svenska. I:
Språkvård 1989:1.
- Karlsson, A., 2009. *Webbplatsöversättning med fokus på användbarhet*. Göteborgs universitet. Hämtad från: <http://hdl.handle.net/2077/18941> [Åtkomstdatum: 2 mars 2009].
- Karlsson, O., 1998. Varför printa när du kan skriva ut? – om Svenska datatermgruppens arbete. I: *Språkvård* 1998:4.
- Karlsson, O., 1997. *Web site, sajt eller webbplats?* I: Josephson, O., (red.), *Svenskan i IT-samhället*. Ord och stil. Uppsala: Hallgren & Fallgren.

- Lemhagen, G., 1991. Hjälpmedel för översättare – en översikt. I: *Språkvård* 1991:1.
- List of Wikipedias – Meta (2009). Wikimedia. Hämtad från: http://meta.wikimedia.org/wiki/List_of_Wikipedias [Åtkomstdatum 20 april 2009].
- Mattsson, B., 2003. De goda, de onda, de fula – och de nya. I: *Språkbruk* 2003:3.
- Nida, E.A., 1964. *Towards a Science of Translating*. Leiden: E.J. Brill.
- Nielsen, J., 2001. *Användbar webbdesign*. Stockholm: Liber.
- Orski, M., 2007. *Öppen källkod i Sverige: beprövad teknik och mediahype*. Stockholm: Obok.
- Prior, M., 2008. HOWTO: Translating OmegaT into your language. *OmegaT*. Hämtad från: http://www.omegat.org/en/howtos/localizing_omegat.html [Åtkomstdatum 26 mars 2009].
- Prior, M., 2009. About OmegaT?, e-brev, 23 maj 2009.
- Raymond, E.S., 2001. *The cathedral and the bazaar [Elektronisk resurs] : musings on Linux and open source by an accidental revolutionary*. Sebastopol: O'Reilly. Hämtad från: <http://www.catb.org/~esr/writings/cathedral-bazaar> [Åtkomstdatum 17 mars 2009].
- Sasikumar, M. m.fl., 2005. *Free/Open Source Software – Guide to Localisation*. Mumbai: Centre for Development of Advanced Computing. Hämtad från: <http://www.iosn.net/l10n/l10n-howto-toolkit/guide.pdf> [Åtkomstdatum 20 mars 2009].
- Souphavanh, A. & Karoonboonyanan, T., 2005. *Free/Open Source Software: Localization*. New Dehli: United Nations Development Programme-Asia Pacific Development. Hämtad från: <http://www.iosn.net/l10n/foss-localization-primer/foss-localization-primer.pdf> [Åtkomstdatum 20 mars 2009].
- Svenska datatermgruppens Ordlista, 2007. Version 27. Svenska datatermgruppen. Hämtad från: <http://www.nada.kth.se/dataterm/rek.html> [Åtkomstdatum 28 april 2009].
- Svenska skrivregler, 2000. Svenska språknämnden. Stockholm: Liber.
- Wolf, D. & Wolf, A., 2001. The Easter Egg Archive FAQ. *The Easter Egg Archive*. Hämtad från: <http://www.eeggs.com/faq.html> [Åtkomstdatum 26 april 2009].